



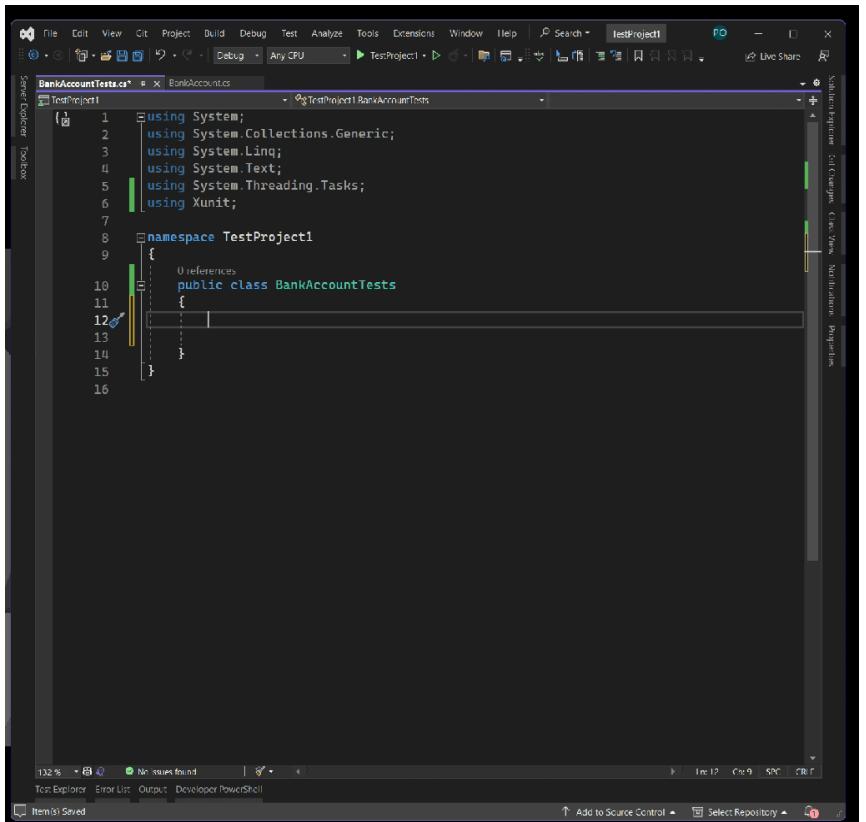
上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

自回归-扩散混合模型

邓志杰

上海交通大学

自回归模型 (AR models) : 从语言生成到智能体Agent, 激发广泛应用



A screenshot of the Microsoft Visual Studio IDE. The main window shows a C# code file named 'BankAccountTests.cs' with the following content:

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using Xunit;
7
8  namespace TestProject1
9  {
10    public class BankAccountTests
11    {
12    }
13}
14
15
16

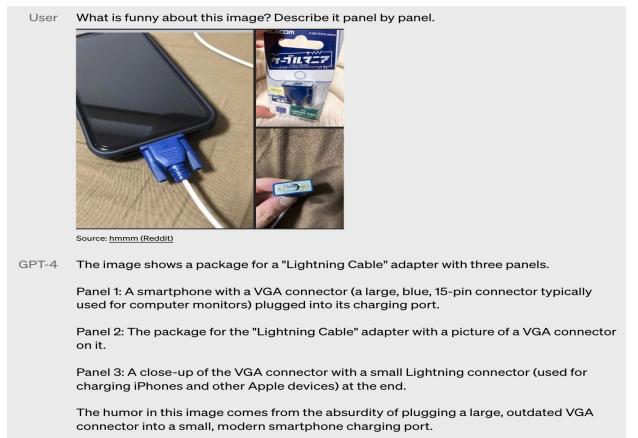
```

The interface includes a toolbar at the top, a menu bar, and various tool windows like 'Server Explorer' and 'Toolbox' on the left.

Coding assistant



Software development



Multimodal understanding

Volume 645 Issue 8081, 18 September 2025



Subscribe

Self-help

Large language models (LLMs) tend to be better at solving problems if they can be trained to set out the steps they take as they try to reach the solution. This kind of 'reasoning' is similar to how humans approach more complex problems, but it presents significant challenges for artificial intelligence, requiring human intervention to add labels and annotations. In this week's issue, researchers at DeepSeek reveal how they trained a model not only to reason in this way but also to do so with minimal human input. The model DeepSeek-R1 was trained using reinforcement learning in which the model was rewarded with a high score when it solved mathematical problems correctly and penalized when it got the answer wrong. As a result, it learnt that reasoning — tackling the problem in a stepwise manner and revealing those steps — was more likely to lead to the correct answer. This led DeepSeek-R1 to self-verify and self-reflect, checking its performance before giving answers to new questions and thereby improving its performance in coding and graduate-level science problems. — [show all](#)

Cover image: Nik Spencer/ Nature.

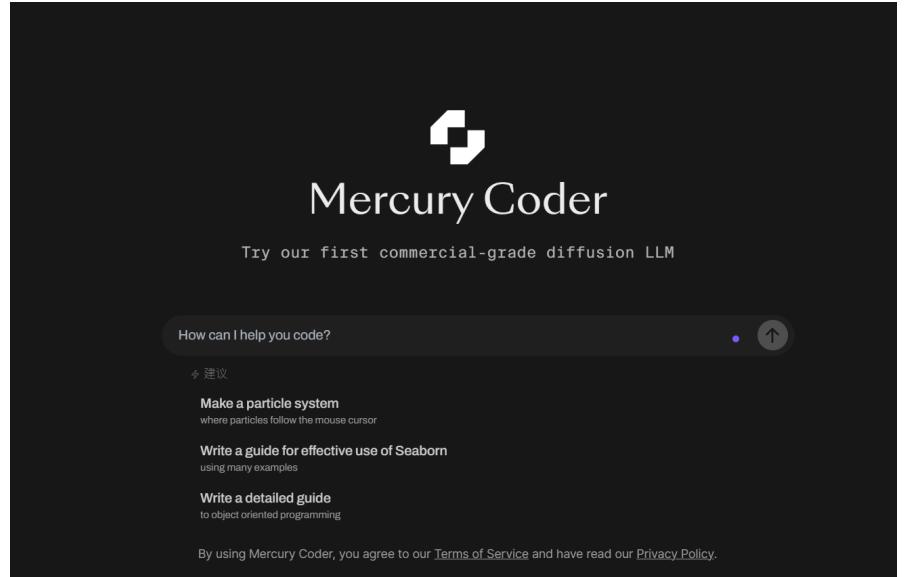
Reasoning

扩散模型 (diffusion models) : 兼顾连续/离散信号建模



Sora by OpenAI

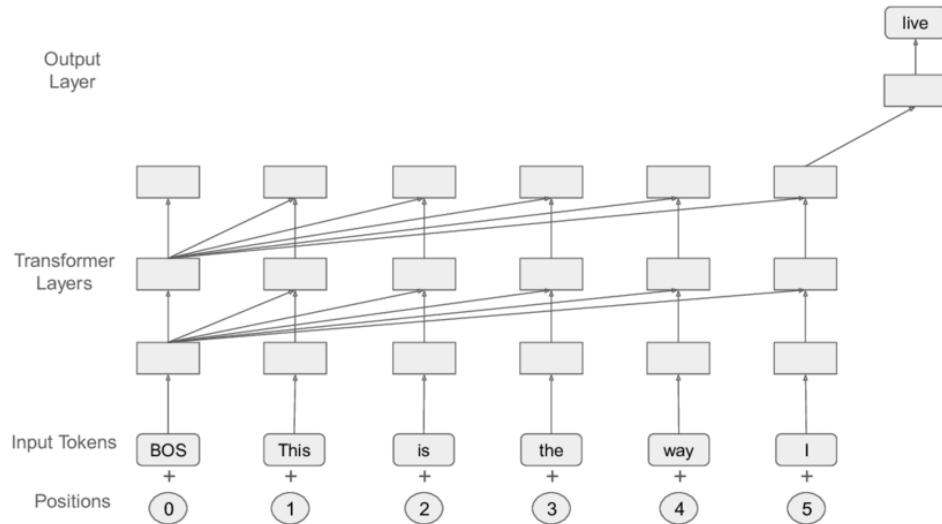
生成长视频等连续视觉信号



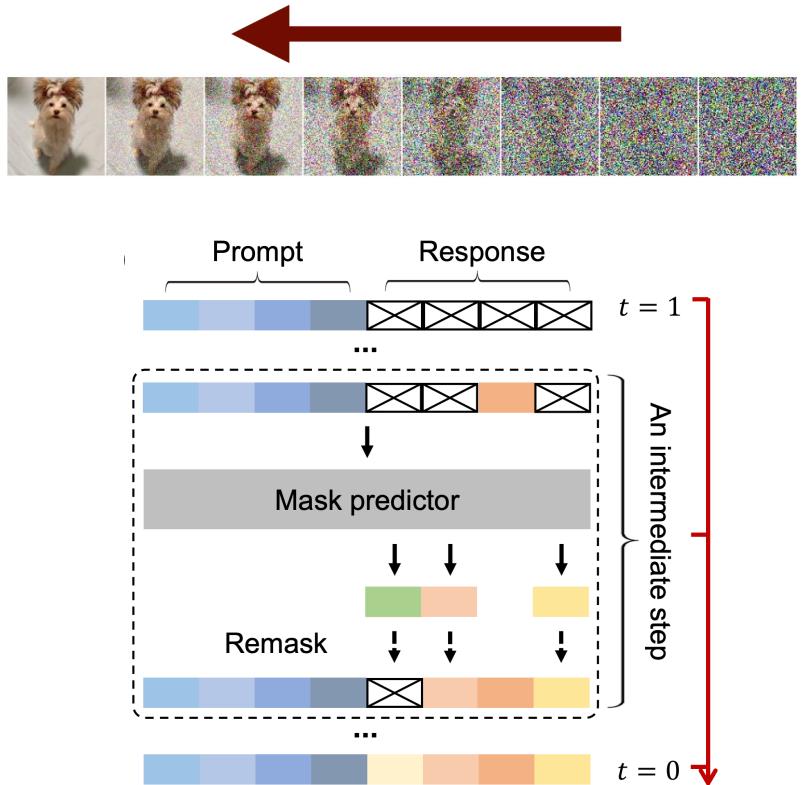
Mercury Coder by Inception Labs

比同性能自回归语言模型快5-10倍

AR和Diffusion各自具有不同的特性



VS.

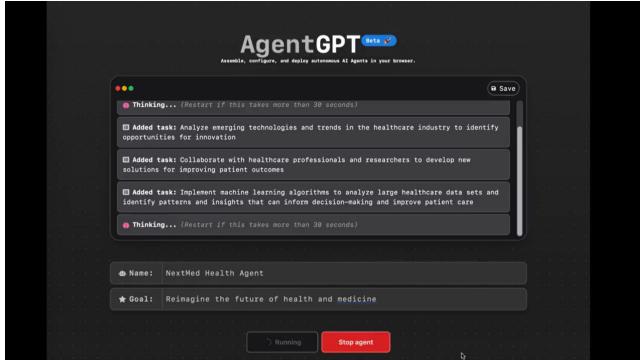


- **AR:** 离散
 - 因果注意力
 - 每一步预测下一个token: $p(x_{-t}|x_{-t'})$

- **Diffusion:** 离散/连续
 - 双向注意力
 - 每一步预测整条数据: $p(x_0|x_t)$

未来大模型架构

AR or diffusion?



Agent (复杂任务、跨模态)



VLA (实时性、强泛化)

AR模型可以提供：世界知识、推理规划、人机交互、研究基础、开源社区、etc.

- 以上场景对于计算高效、扩模态高效、数据高效等方面有很高的要求，然而
 - AR是计算效率、跨模态效率和数据效率最优吗？

未来大模型架构

纯AR存在计算效率、扩模态效率等方面的问题

AR模型顺序推理（逐token生成），
低效昂贵

AR模型依赖离散token，但离散化产
生信息损失

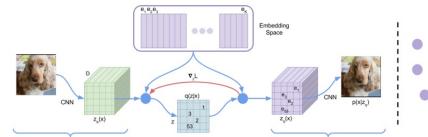
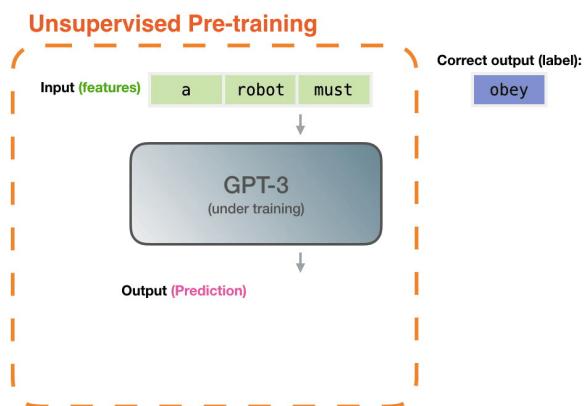


Figure 1: Left: A figure describing the VQ-VAE. Right: Visualisation of the embedding space. The output of the encoder $z(x)$ is mapped to the nearest point e_2 . The gradient $\nabla_z L$ (in red) will push the encoder to change its output, which could alter the configuration in the next forward pass.





解决方案：自回归-扩散混合模型 (AR-Diffusion Hybrid Models)

从解决AR模型的问题的角度

AR+discrete diffusion

- 动机: diffusion每一步都进行全局推理

将AR语言模型生成tokens数和推理步数解耦
(高效并行解码)

AR+continuous diffusion

- 动机: diffusion可以直接生成连续信号

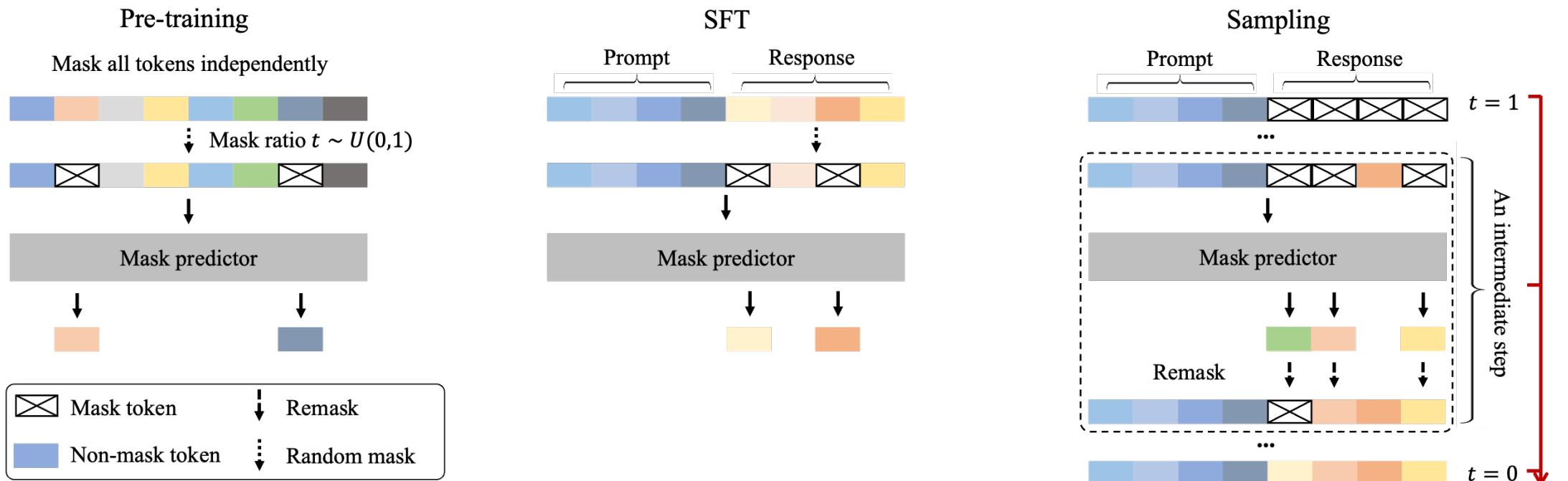
为AR语言模型提供生成多模态内容的能力
(跨模态生成)



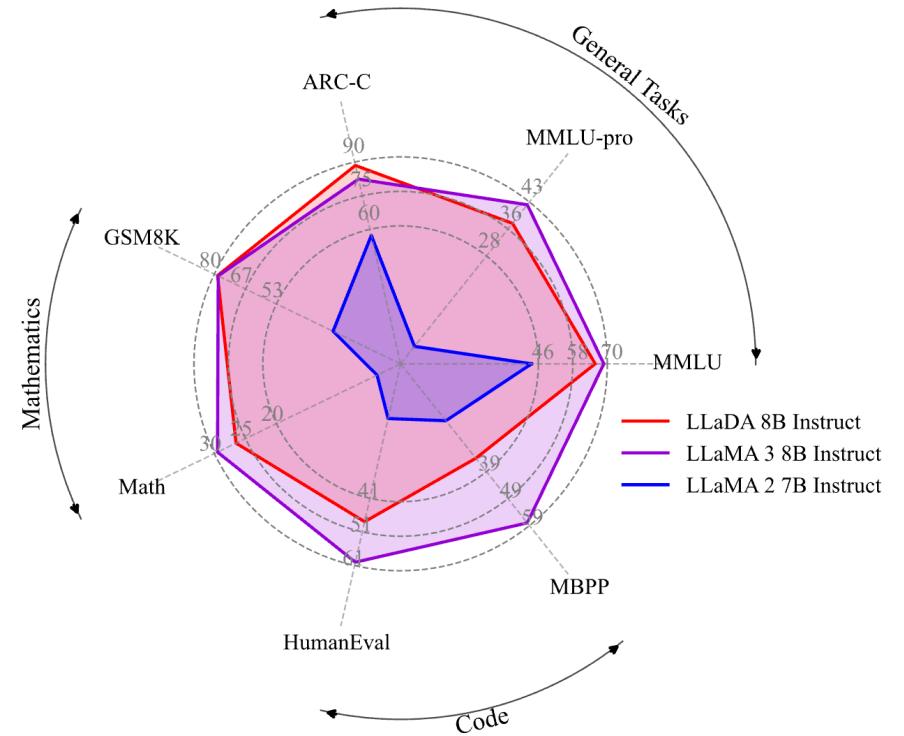
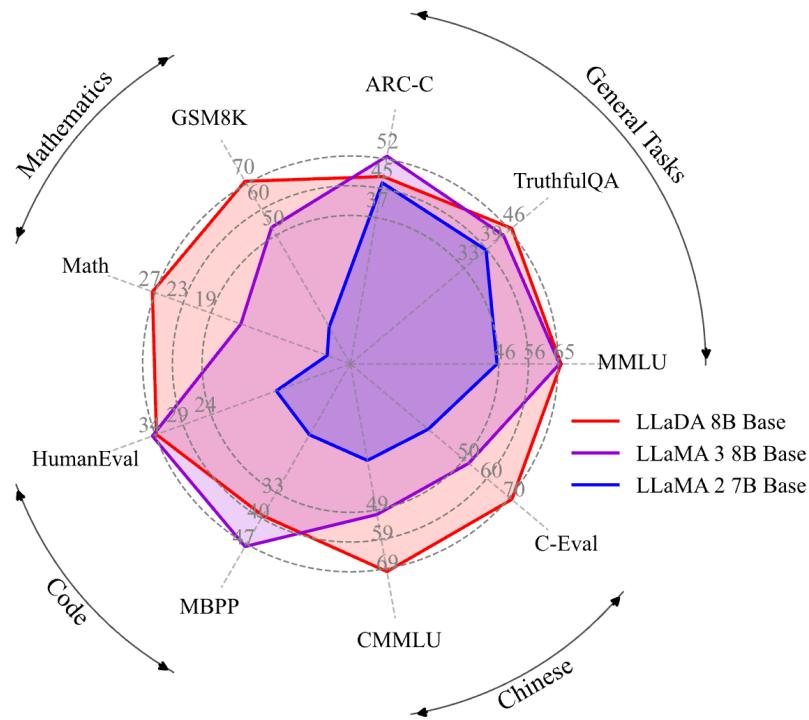
自回归-扩散混合的高效推理语言模型

扩散语言模型

离散扩散模型 (discrete diffusion models)



对扩散语言模型的期待：并行生成、推理加速

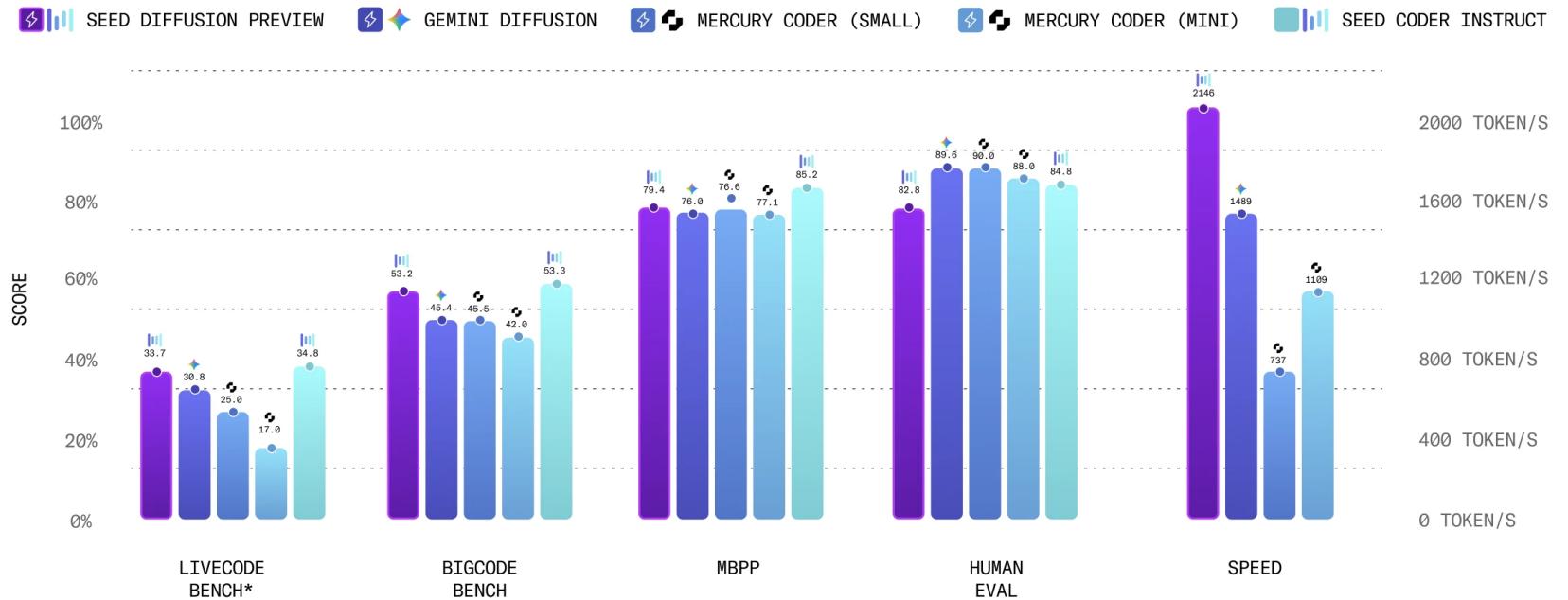


LLaDA (The first Large Language Diffusion Model, RUC), 可达LLaMA3级水平

对扩散语言模型的期待：并行生成、推理加速

SEED DIFFUSION: A LARGE-SCALE
DIFFUSION LANGUAGE MODEL WITH HIGH-
SPEED INFERENCE

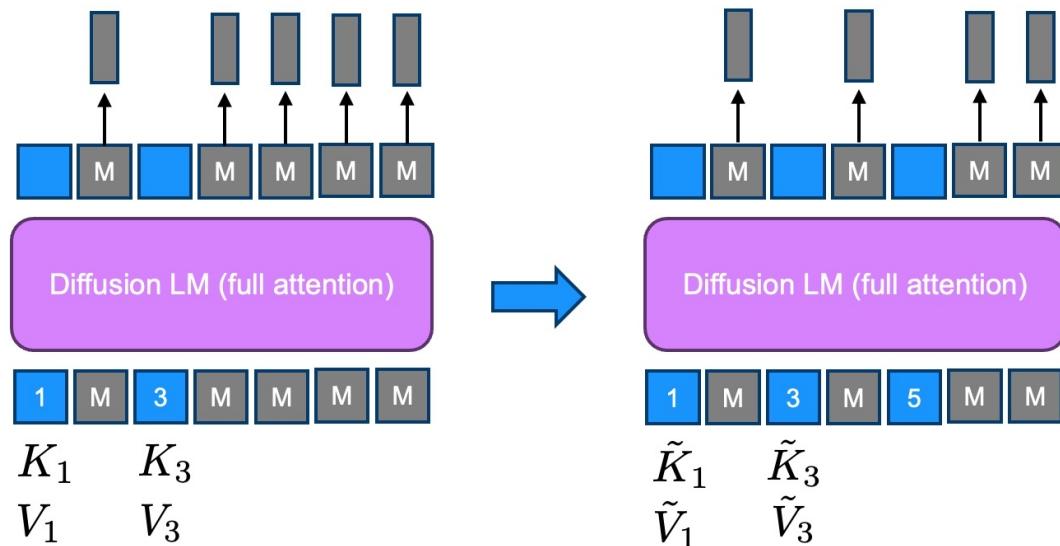
RESOURCE: [ARXIV.ORG](https://arxiv.org/)



Seed Diffusion Preview: A large scale language model based on discrete-state diffusion, specializing in code generation, achieves an inference speed of 2,146 token/s, a 5.4x improvement over autoregressive models of comparable size.

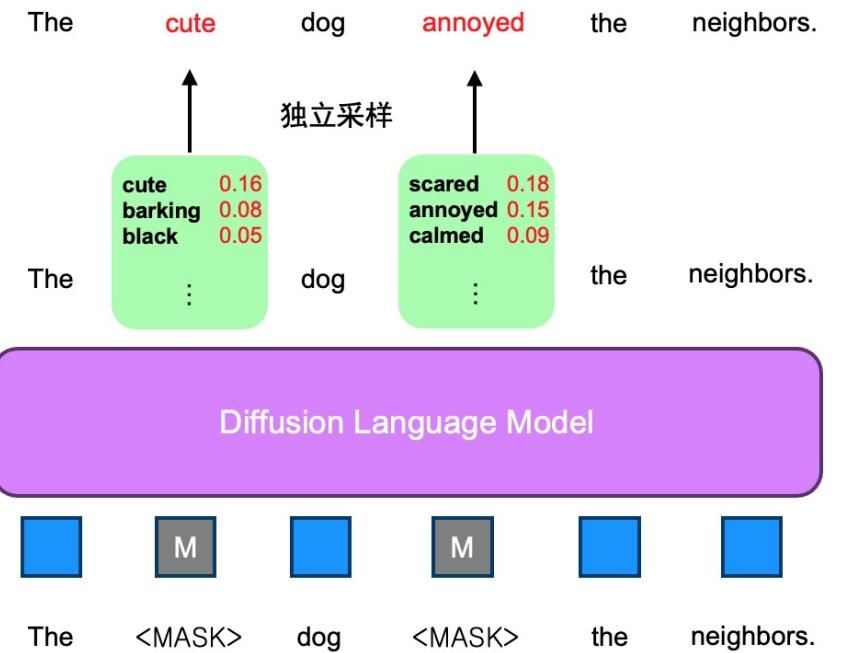
然而，对扩散语言模型的期待在开源社区并未兑现

- 无法应用KV cache：已经被解码的token的key和value始终会在去噪过程中发生变化
- 同时解码多个token时会产生潜在的训练-推理不一致的风险



由于扩散模型的全局注意力机制，已经被解码的token的key和value依旧会在生成过程中产生变化，每一步迭代都需重新计算

- 同时解码多个token时会产生潜在的训练-推理不一致的风险



虽然diffusion LM可以一次预测多个token的logits，其采样相互独立，同时解码相关性强的多个token可能会产生冲突



自回归语言模型 vs 扩散语言模型

due

that

Diffusion LLM

I love LLM <m> to <m>

✓ 有并行潜力

✗ 不能用KV Cache

✗ 固定长度

✗ 目前性能稍逊

due

to

the

AR LLM

I love LLM due to that

✗ 串行生成

✓ 可以用KV Cache

✓ 任意长度

✓ 性能更强

构建自回归-扩散混合的语言高效生成新范式？



自回归-扩散混合的语言生成

Autoregression:

✓ High quality ✓ Arbitrary-length ✓ KV caching

Generation steps

There are three categories of the average
There are three categories of the average rate
There are three categories of the average rate of...

✗ Not Parallelizable

Diffusion:

✗ Lower quality ✗ Fixed-length ✗ No KV caching

✓ Parallelizable

the reusability will continue to the
Repeal the reusability cuts and the law will continue to reduce the
Repeal the reusability cuts and prove the law will continue to reduce the deficit.

Block Diffusion (Ours):

✓ High quality ✓ Arbitrary-length ✓ KV caching

✓ Parallelizable

On September 17, we be
On September 17, 2016, we will be giving the release of
On September 17, 2016, we will be giving the beta-release of the to our server testing ...

Block diffusion

- teacher forcing

分块的语言生成：

✓ 块间自回归：复用KV cache

✓ 块内扩散：并行生成

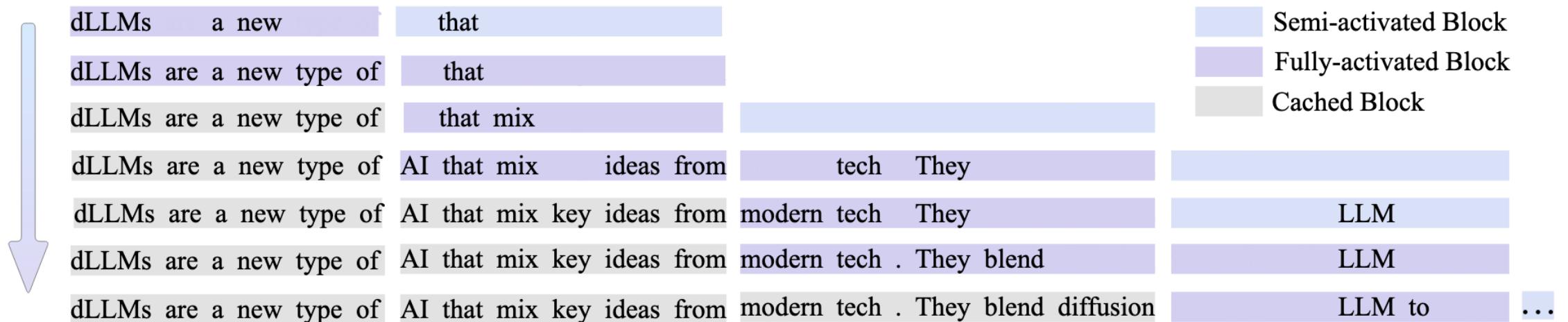
✓ 任意生成长度

✗ 然而，完全杜绝了块间并行



自回归-扩散混合的语言模型

Discrete Diffusion Forcing (D2F): block-wise顺序生成 + inter-block 并行



分块的语言生成：

✓ 块间自回归：复用KV cache

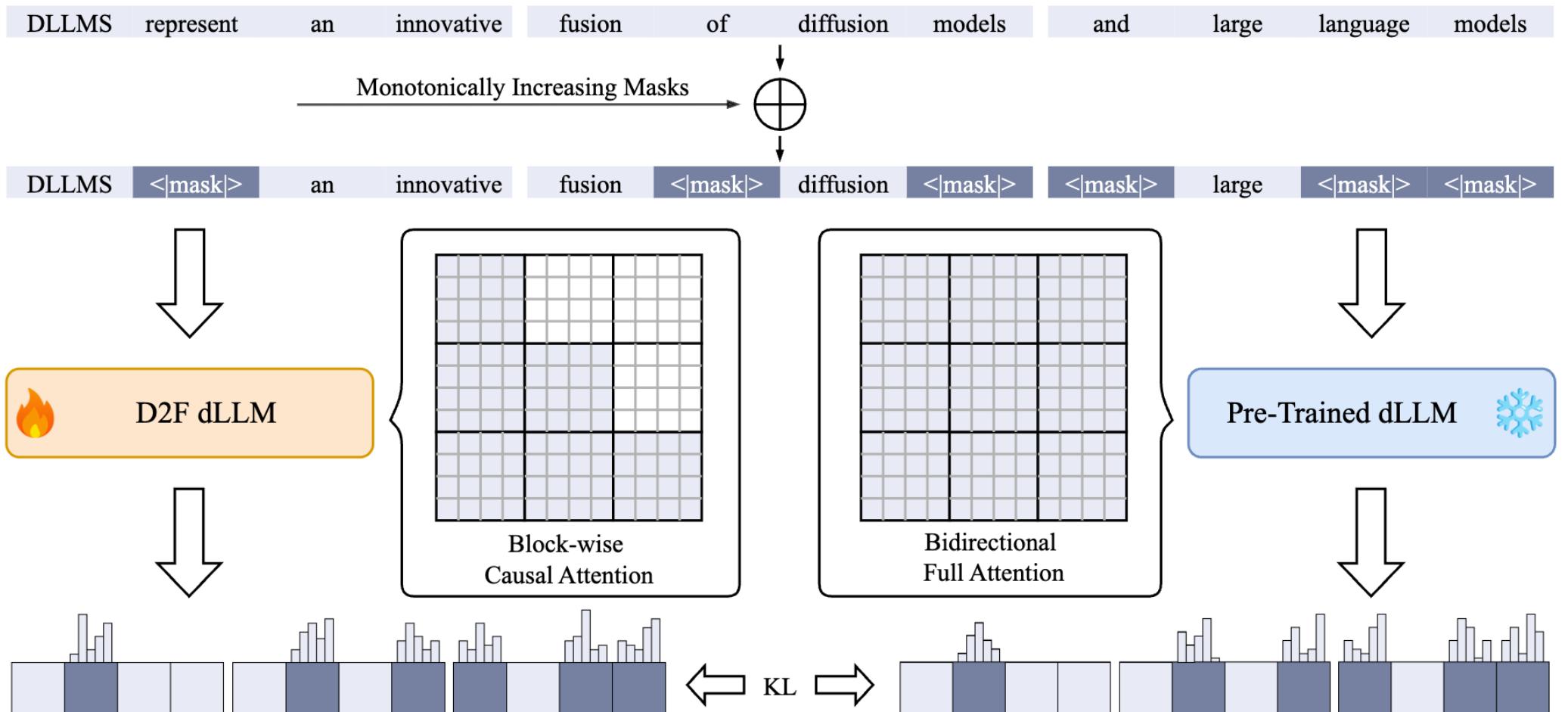
✓ 块内扩散：并行生成

✓ 任意生成长度

✓ 允许块间并行生成

自回归-扩散混合的语言模型

Discrete Diffusion Forcing (D2F): 通过非对称蒸馏进行训练





自回归-扩散混合的语言模型

Discrete Diffusion Forcing (D2F): 效果

| Test Set | Method | TPS ↑ | Latency (s) ↓ | Gen. Length | Score ↑ |
|---------------------|--------------------------|---------------------|--------------------|-------------|-------------|
| GSM8K 4-shot | LLaDA-Instruct | 7.2 (1.0x) | 32.3 (1.0x) | 231 | 77.4 |
| | dLLM-Cache | 20.1 (2.8x) | 11.5 (2.8x) | 231 | 77.5 |
| | Fast-dLLM (Prefix-Cache) | 33.3 (4.6x) | 7.0 (4.6x) | 232 | 77.8 |
| | Fast-dLLM (Dual-Cache) | 35.2 (4.9x) | 6.6 (4.9x) | 232 | 78.9 |
| | D2F-LLaDA | 52.5 (7.3x) | 2.8 (11.5x) | 144 | 77.3 |
| MBPP 3-shot | LLaDA-Instruct | 0.9 (1.0x) | 71.4 (1.0x) | 65 | 39.0 |
| | dLLM-Cache | 2.3 (2.6x) | 28.3 (2.5x) | 66 | 37.0 |
| | Fast-dLLM (Prefix-Cache) | 13.0 (14.4x) | 4.9 (14.6x) | 64 | 37.6 |
| | Fast-dLLM (Dual-Cache) | 15.3 (17.0x) | 3.8 (18.8x) | 58 | 36.4 |
| | D2F-LLaDA | 47.6 (52.9x) | 1.4 (51.0x) | 68 | 38.0 |
| HumanEval 0-shot | LLaDA-Instruct | 2.8 (1.0x) | 38.8 (1.0x) | 107 | 36.0 |
| | dLLM-Cache | 4.5 (1.6x) | 23.3 (1.7x) | 104 | 39.0 |
| | Fast-dLLM (Prefix-Cache) | 13.7 (4.9x) | 7.4 (5.2x) | 102 | 38.4 |
| | Fast-dLLM (Dual-Cache) | 19.2 (6.9x) | 5.2 (7.5x) | 100 | 35.4 |
| | D2F-LLaDA | 81.6 (29.1x) | 1.6 (24.3x) | 133 | 40.2 |
| Math 4-shot | LLaDA-Instruct | 21.1 (1.0x) | 11.5 (1.0x) | 243 | 23.7 |
| | dLLM-Cache | 26.9 (1.3x) | 9.1 (1.3x) | 246 | 23.2 |
| | Fast-dLLM (Prefix-Cache) | 47.7 (2.3x) | 5.2 (2.2x) | 246 | 22.4 |
| | Fast-dLLM (Dual-Cache) | 42.5 (2.0x) | 5.8 (2.0x) | 246 | 22.4 |
| | D2F-LLaDA | 90.2 (4.3x) | 4.3 (2.7x) | 384 | 29.1 |

Table 1: Performance comparison of various acceleration methods on the **LLaDA-Instruct** model. Speedup ratios relative to the baseline are shown in (green). All baseline methods use the default sampling configuration from the original LLaDA implementation.

相对于原始的DLMs，在代码生成上
达到超过50倍的加速



自回归-扩散混合的语言模型

Discrete Diffusion Forcing (D2F): demo

D2F-LLaDA-Instruct-8B  LLaMA3-Instruct-8B



自回归-扩散混合的语言模型

Discrete Diffusion Forcing (D2F): demo

❓ Enter your question

Solve the equation $x^2 - \underline{6x} + 8 = 0$.

自回归-扩散混合的语言模型

Discrete Diffusion Forcing (D2F): 第一个大幅超越AR模型速度的开源扩散语言模型

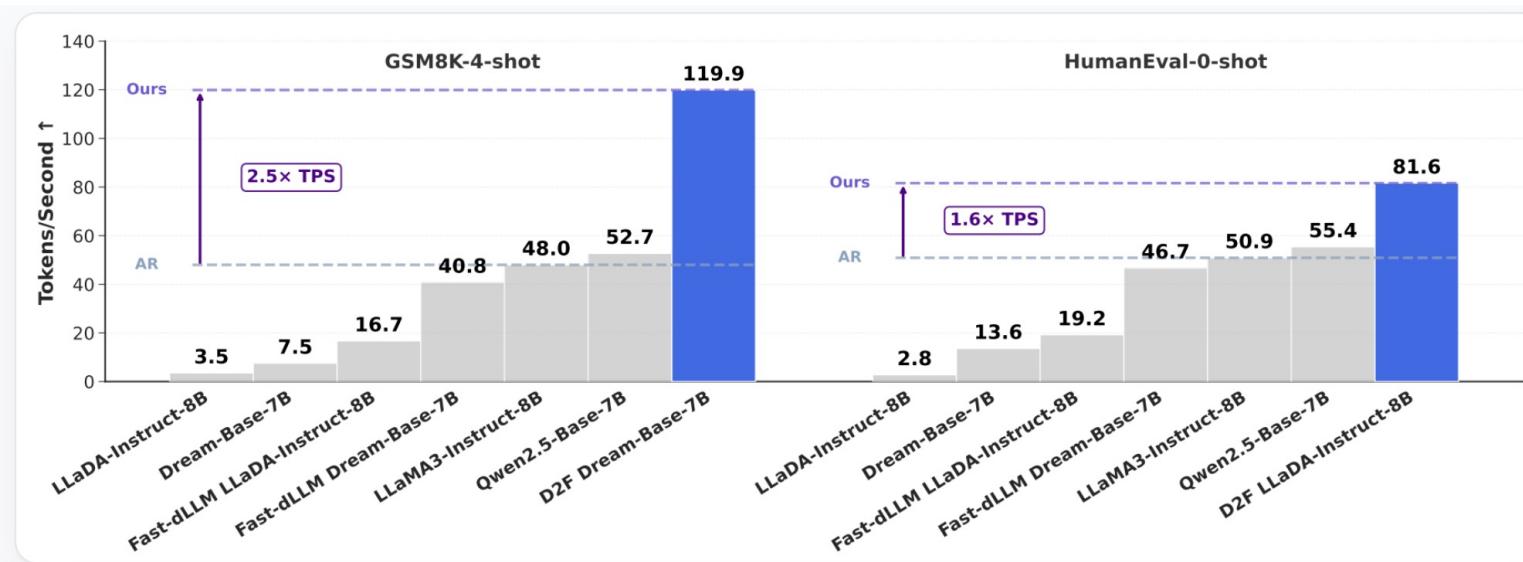
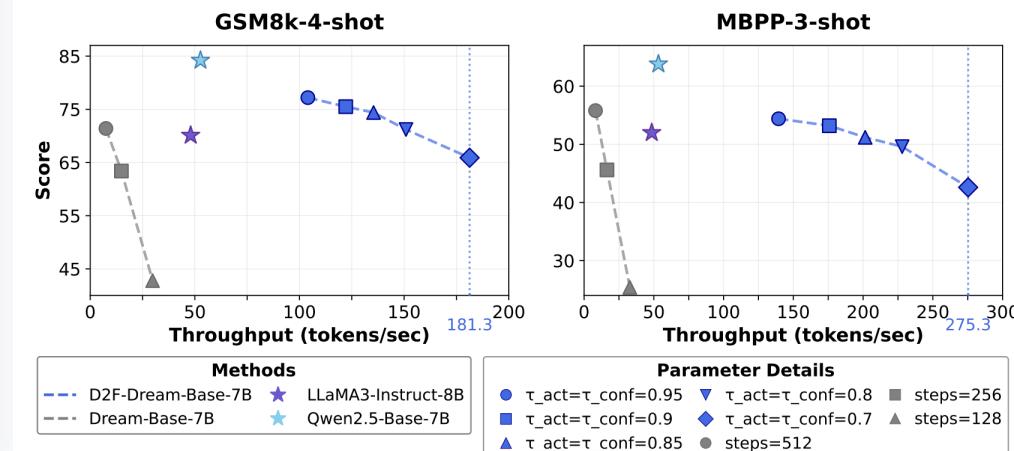


Figure 1. D2F dLLMs surpass similarly-sized autoregressive (AR) models in inference speed for the first time, achieving up to **2.5x** higher throughput than LLaMA3 and Qwen2.5. This comparison includes vanilla dLLMs, previous acceleration methods, and AR baselines.



提供了宝贵的Throughput vs.
performance trade-off



自回归-扩散混合的语言模型

推理引擎 **D2F-Engine**: 基于Nano-vLLM实现专有适配

| | Method | TPS (Throughput) | Score |
|-----------|------------------|-----------------------------|-------|
| HumanEval | HF Backend | 70.37 | 47.56 |
| | D2fEngine | 716.31 (179.08 x 4) | 50.61 |
| GSM8K-CoT | HF Backend | 93.71 | 78.17 |
| | D2fEngine | 1182.76 (295.69 x 4) | 75.59 |

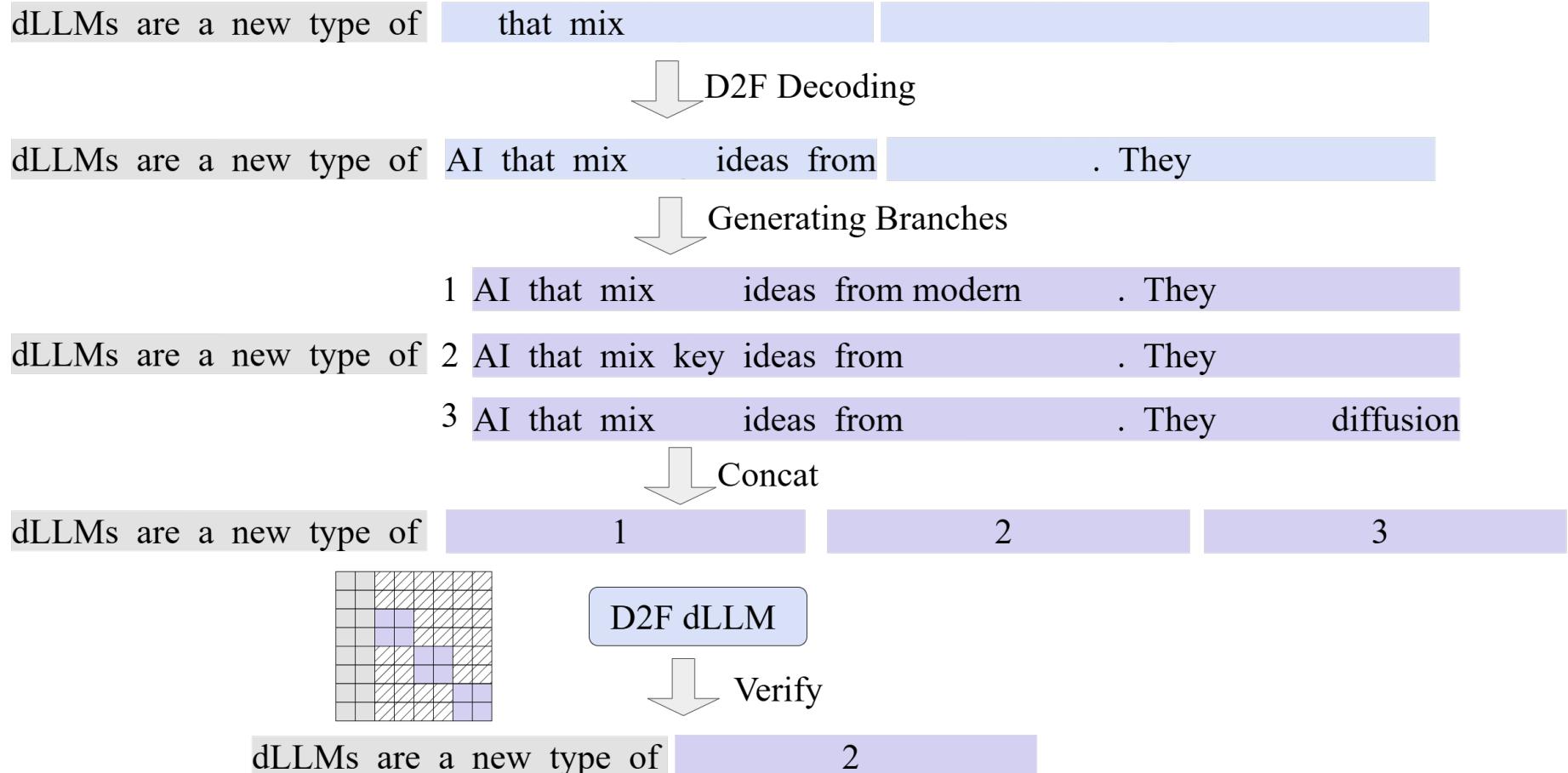
| Parameter | Value |
|-------------------------|---------|
| enforce_eager | True |
| data_parallel_size | 4 |
| tensor_parallel_size | 1 |
| gpu_memory_utilization | 0.3 |
| max_num_batched_tokens | 5120 |
| max_num_seqs | 64 |
| max_model_len | 5120 |
| complete_threshold | 0.95 |
| add_new_block_threshold | 0.90 |
| kv_cache_layout | unified |

Model: Dream-v0-Base-7B; Devices: 4 * H20-NVLink 96G



解码速度如何随计算量做scale?

- batch-size=1, D2F平均每个step可以解码3.x个token
- 参考LLM的推测解码：多个proposals，并行验证





解码速度如何随计算量做scale? D2F的推测解码

- batch-size=1, D2F平均每个step可以解码3.x个token
- D2F 推测解码: 平均每个step解码 >7 个token, 同时平均性能几乎不下降

| gsm8k | tokens/steps | score |
|----------|--------------|-------|
| D2F | 3.3 | 77.9 |
| D2F+并行验证 | 7.7 | 76.3 |

| mbpp | tokens/steps | score |
|----------|--------------|-------|
| D2F | 2.8 | 37.6 |
| D2F+并行验证 | 5.1 | 38.6 |

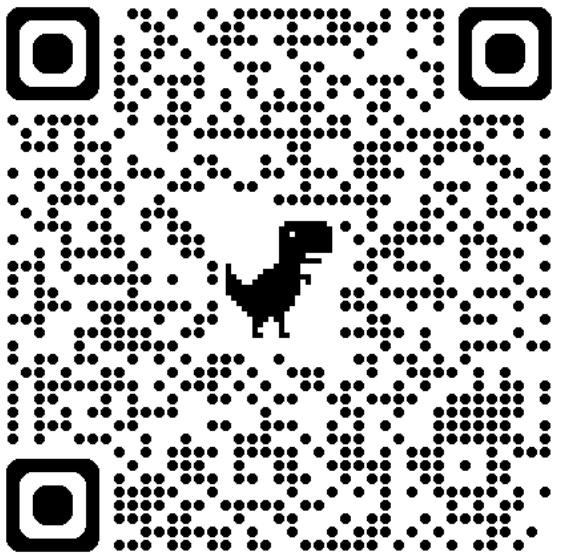
| humaneval | tokens/steps | score |
|-----------|--------------|-------|
| D2F | 3.5 | 40.2 |
| D2F+并行验证 | 6.1 | 38.4 |

| math | tokens/steps | score |
|----------|--------------|-------|
| D2F | 3.2 | 29.1 |
| D2F+并行验证 | 7.2 | 30.7 |

Model: LLaDA-Instruct-7B



Discrete Diffusion Forcing (D2F): 开源



GitHub

<https://github.com/zhijie-group/Discrete-Diffusion-Forcing>



Engine

https://github.com/zhijie-group/d2f_vllm



自回归-扩散混合的多模态内容生成



“统一生成模型”

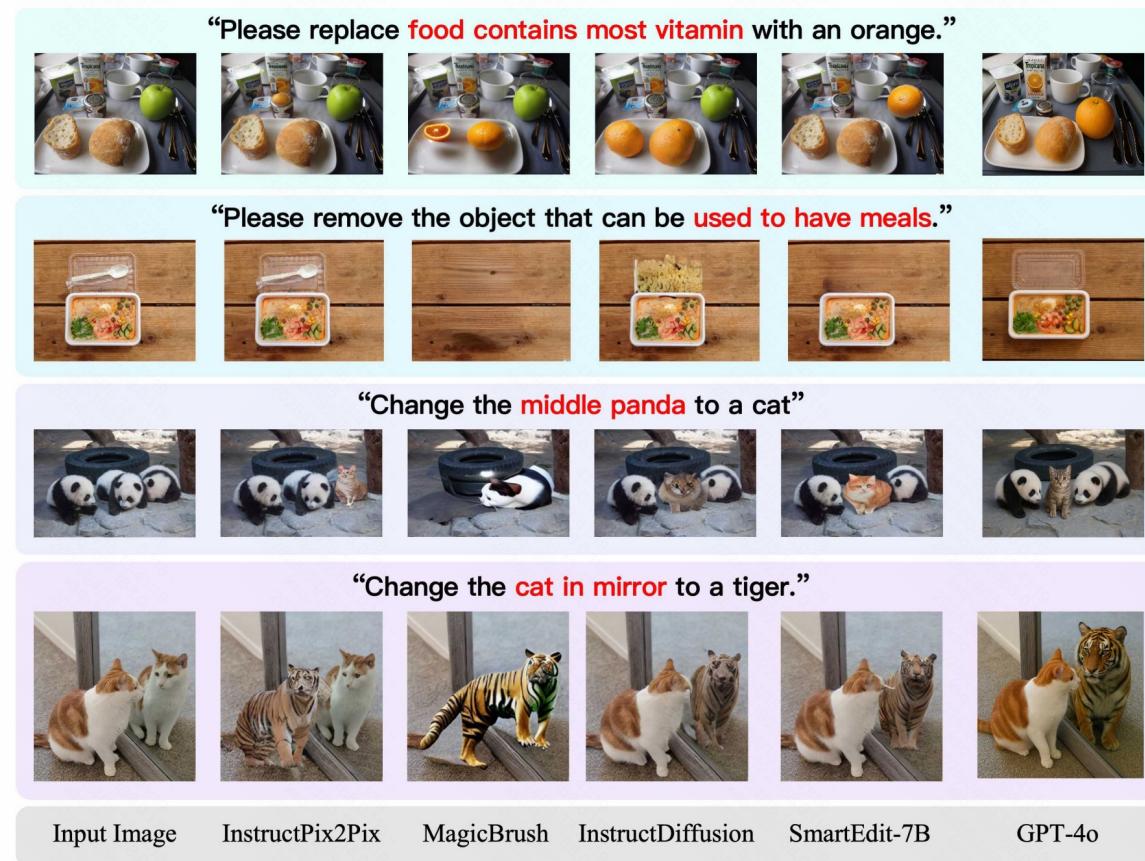
基于任意组合的**condition**，生成多模态内容（如：图像、文本、视频、声音、...）



动机：GPT-4o代表的“统一生成模型”（面向图文）

相较于专用的图像生成模型，统一语言和视觉建模有助于建立世界知识

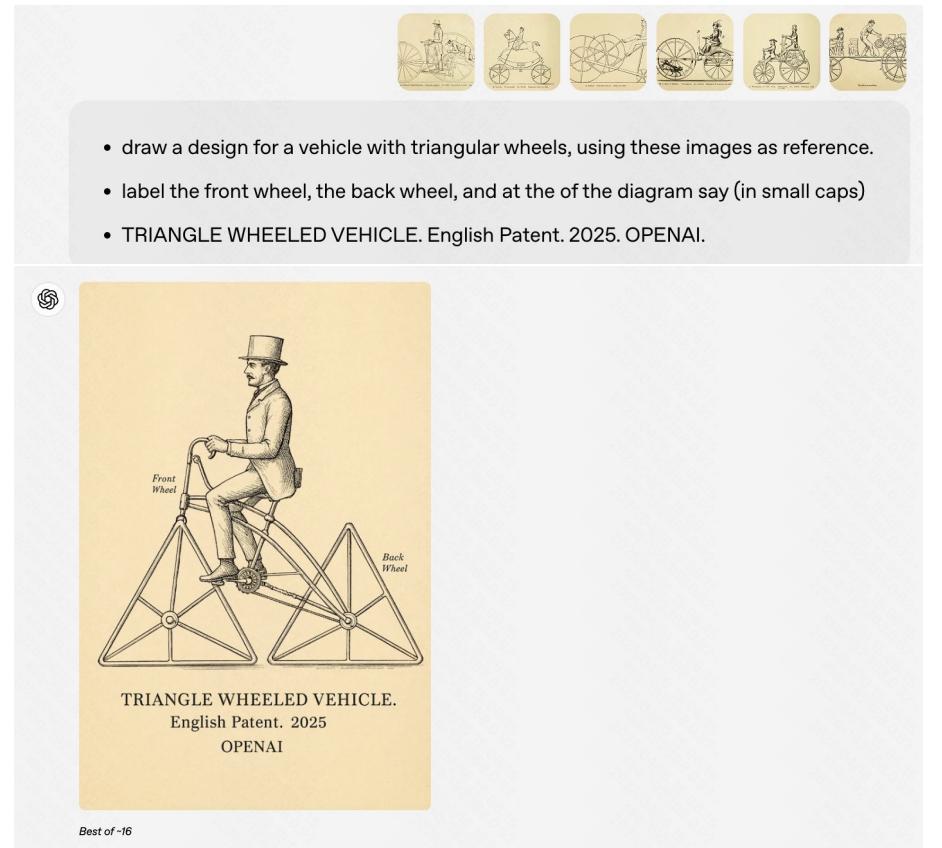
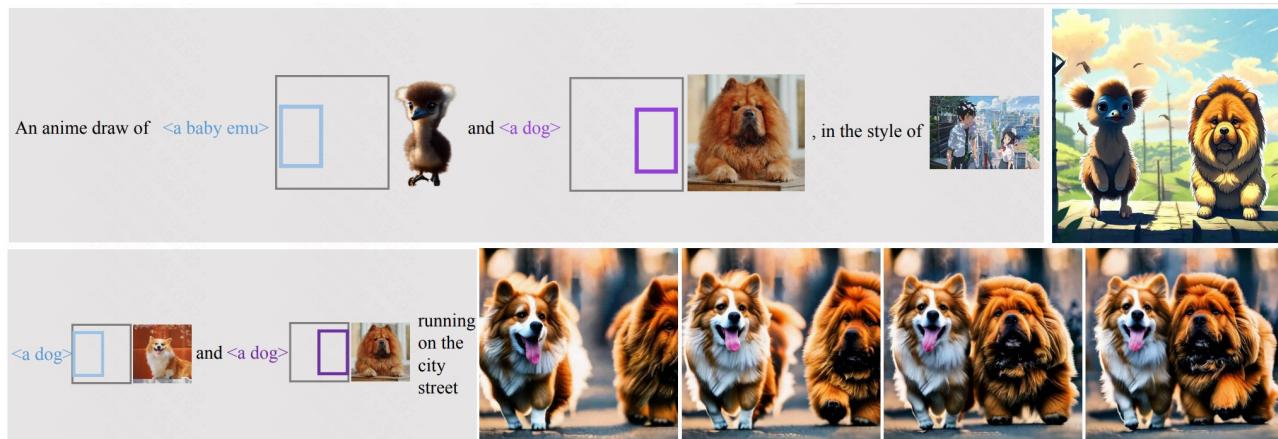
- 在传统编辑任务中的指令理解与跟随能力显著增强



动机：GPT-4o代表的“统一生成模型”（面向图文）

天然具备长上下文学习能力

- 处理多图与文本混合输入时，统一模型能够有效整合多模态信息，展现了控制精准、主体一致性的生成效果



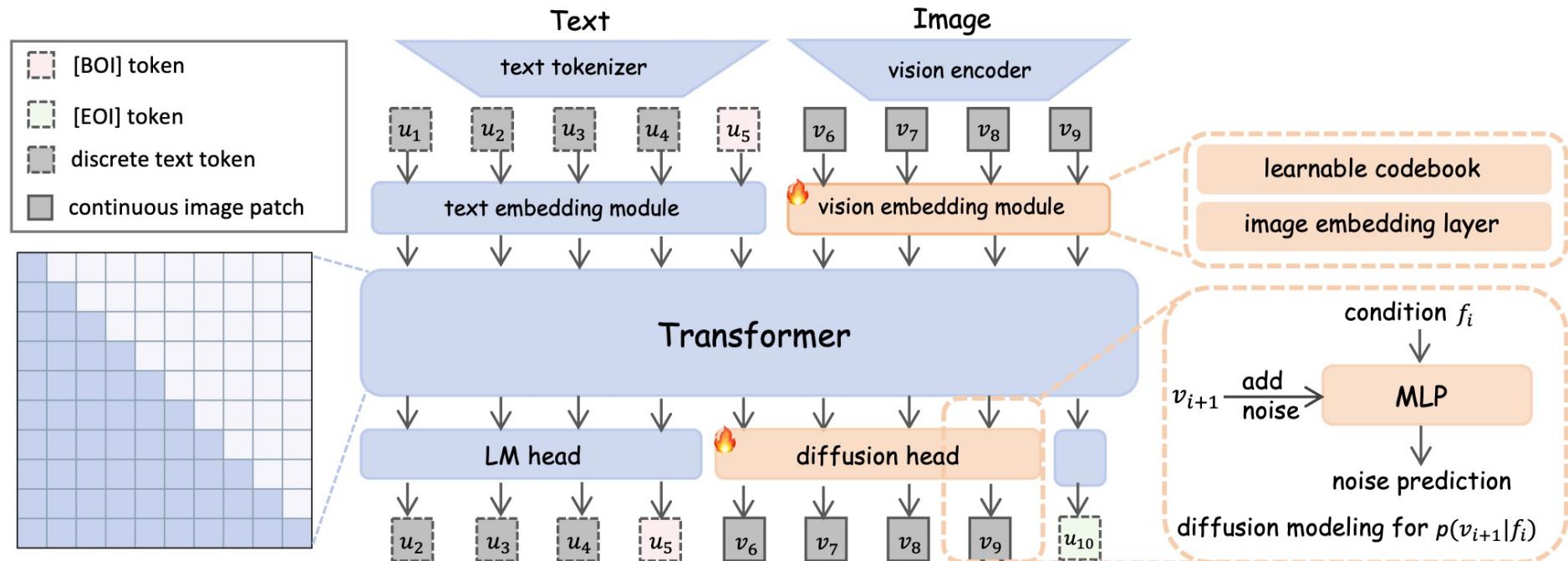


如何构建“统一生成模型”

AR-diffusion 混合模型是可行路径，也是近期趋势，但AR和diffusion分别起什么作用？

- AR来“统一”，建立全局关联
 - 文：预测下一个离散token
 - 图：预测下一个图像单元的语义表示
- Diffusion来“适配”，链接各种连续模态
 - 图：以语义表示为条件生成连续的pixels

Orthus: 自回归主干+扩散头



- **自回归Transformer主干 (拥抱KV Cache)**
- **处理离散的文本token和连续的图像feature (基于连续VAE)**
- **基于线性层定义的language head和diffusion MLP来分别生成文和图 (逐token/patch)**

Orthus: 自回归主干+扩散头

- 从离散图像特征到连续特征:

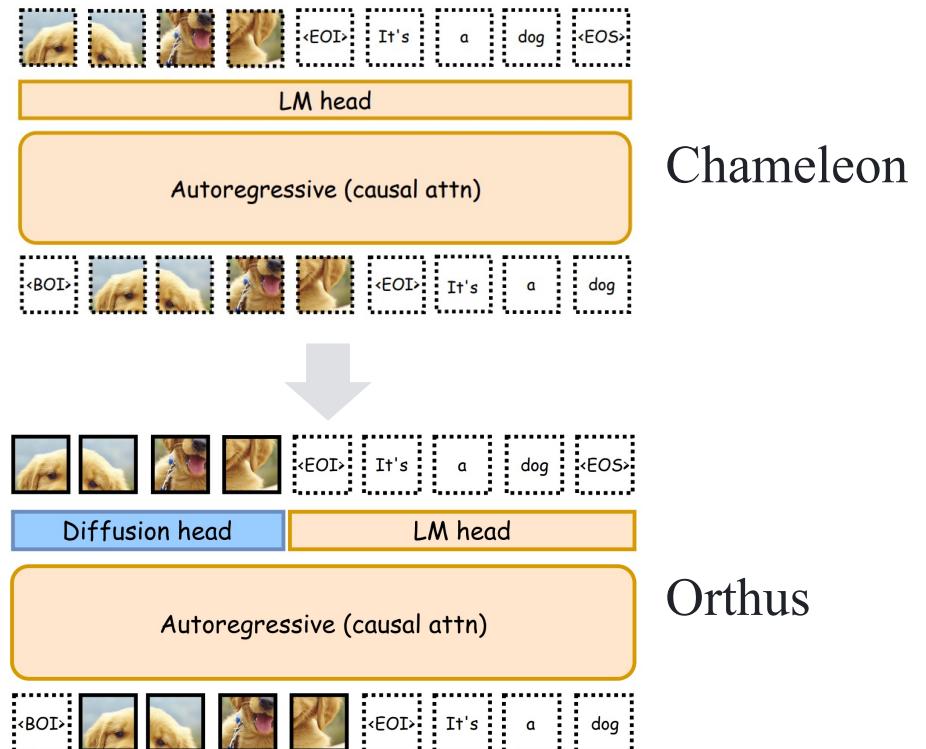
$$h_i = \sum_j w_j \mathbb{1}_{\tilde{v}_i=j}, \tilde{v}_i = \arg \min_{j \in \{1, \dots, K\}} d(\mathbf{v}_i, \mathbf{c}_j)$$

=>

$$h_i = \sum_j w_j \frac{e^{-d(\mathbf{v}_i, \mathbf{c}_j)/\tau}}{\sum_{k=1}^K e^{-d(\mathbf{v}_i, \mathbf{c}_k)/\tau}}$$

- 自回归统一模型 (如: Chameleon) : $\tau=0$
- Orthus: $\tau=1$**

- 从 $\tau=0$ 的模型冷启动
 - 72个A100 GPU hours即可得到Orthus-7B-base
 - 将涉及的VQ-VAE调成了VAE



| Model | PSNR↑ | SSIM [63]↑ |
|--------|-------|------------|
| VQ-VAE | 23.7 | 0.80 |
| Ours | 26.1 | 0.84 |

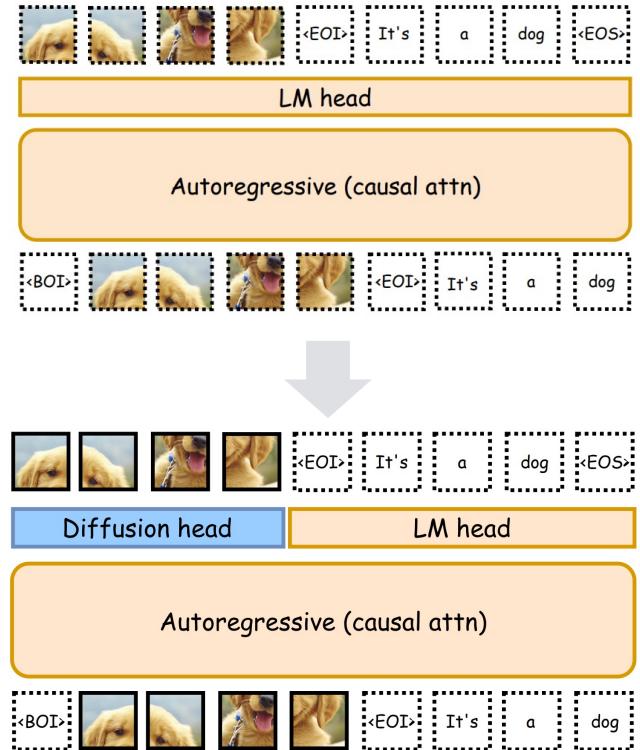
Orthus: 自回归主干+扩散头

- **Diffusion head**训练:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\epsilon, t} [\|\epsilon - \epsilon_{\theta}(\sqrt{\alpha_t} \mathbf{v}_{i+1} + \sqrt{1 - \alpha_t} \epsilon, t, \mathbf{f}_i)\|_2^2]$$

- 从文到图/图到文等不同任务学习有价值信号
 - 1:1混合LlaVA-v1.5-665K指令微调数据和高质量文生图数据JourneyDB、LAION-COCO-aesthetic (recaptioned from ShareGPT-4v)

$$\mathcal{L}_{\text{Orthus}} = \mathcal{L}_{\text{ar}} + \lambda \mathcal{L}_{\text{diff}}$$





Orthus: ablation结果

Table 4. Comparisons of the performance of Orthus via separate training and unified training across multimodal benchmarks.

| Type | $\mathcal{L}_{\text{diff}}$ | \mathcal{L}_{ar} | POPE↑ | MME-P↑ | GQA↑ | GenEval↑ |
|-------------|-----------------------------|---------------------------|-------------|---------------|-------------|-------------|
| Und. only | ✗ | ✓ | 78.7 | 1244.2 | 51.9 | - |
| Gen. only | ✓ | ✗ | - | - | - | 0.56 |
| Und. & Gen. | ✓ | ✓ | 79.6 | 1265.8 | 52.8 | 0.58 |

同时从文到图和图到文数据学习可以实现1+1>2

Table 5. Ablation study on the choice of vision embedding modules on visual understanding tasks.

| Type | POPE↑ | MME-P↑ | VQAv2↑ | GQA↑ | MMMU↑ |
|---------|-------------|---------------|-------------|-------------|-------------|
| softmax | 78.7 | 1244.2 | 60.8 | 51.9 | 28.0 |
| argmin | 77.6 | 1064.8 | 57.9 | 50.1 | 26.7 |
| linear | 70.4 | 800.7 | 50.3 | 44.5 | 22.3 |

连续的图像特征对于视觉理解任务必要，但要避免冷启动



Orthus: 文生图/图生文量化结果

- 在多个图像理解指标上超越了现有混合理解生成模型Chameleon和Show-o，并在文到图生成的 GenEval 指标上超过SDXL

Table 3. Comparison with state-of-the-arts on visual generation benchmarks. Model using external pre-trained diffusion model is marked with * and Chameleon[†] is post-trained with the same dataset as Orthus. The results in **bold** and underline are the best and second-best results, respectively.

| Type | Model | Res. | GenEval | HPS |
|-------------|---------------------------------|------|-------------|-------------|
| | SDv1.5 (Rombach et al., 2022) | 512 | 0.43 | 27.0 |
| | SDv2.1 (Rombach et al., 2022) | 512 | 0.50 | 27.2 |
| Gen. Only | DALL-E (Ramesh et al., 2022) | 512 | 0.52 | 26.9 |
| | Emu3-Gen (Wang et al., 2024) | 512 | 0.54 | - |
| | SDXL (Podell et al., 2023) | 512 | 0.55 | 30.9 |
| | SD3(d=30) (Esser et al., 2024) | 512 | 0.64 | - |
| Und. & Gen. | SEED-X* (Ge et al., 2024) | 448 | 0.49 | - |
| | LWM (Liu et al., 2024e) | 256 | 0.47 | 26.1 |
| | Show-o (Xie et al., 2024) | 256 | 0.53 | 27.3 |
| | Transfusion (Zhou et al., 2024) | 256 | 0.63 | - |
| | Chameleon [†] | 512 | 0.43 | 26.9 |
| | Orthus (Ours) | 512 | <u>0.58</u> | 28.2 |

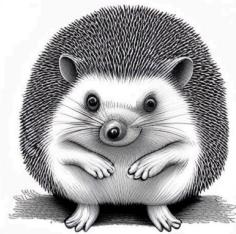
Table 2. Evaluation on visual understanding benchmarks. Und. and Gen. denote “understanding” and “generation”, respectively. Models using external pre-trained diffusion models are marked with * and Chameleon[†] is post-trained with the same dataset as Orthus. The results in **bold** and underline are the best and second-best results, respectively. The results correspond to the exact match accuracy.

| Type | Model | # Params | POPE↑ | MME-P↑ | VQAv2↑ | GQA↑ | MMMU↑ |
|---------------|-----------------------------------|----------|-------------|---------------|-------------|-------------|-------------|
| Und. Only | LlaVa (Liu et al., 2024d) | 7B | 76.3 | 809.6 | - | - | - |
| | LlaVA-v1.5 (Liu et al., 2024b) | 7B | 85.9 | 1510.7 | 78.5 | 62.0 | 35.4 |
| | InstructBLIP (Dai et al., 2023) | 7B | - | - | - | 49.2 | - |
| | Qwen-VL-Chat (Bai et al., 2023) | 7B | - | 1487.5 | 78.2 | 57.5 | - |
| | Emu3-Chat (Wang et al., 2024) | 8B | 85.2 | 1243.8 | 75.1 | 60.3 | 31.6 |
| | InstructBLIP (Dai et al., 2023) | 13B | 78.9 | 1212.8 | - | 49.5 | - |
| Und. and Gen. | Emu* (Sun et al., 2023) | 13B | - | - | 52.0 | - | - |
| | NExT-GPT* (Wu et al., 2013) | 13B | - | - | 66.7 | - | - |
| | Gemini-Nano-1 (Team et al., 2023) | 1.8B | - | - | 62.7 | - | 26.3 |
| | Show-o (Xie et al., 2024) | 1.3B | 73.8 | 948.4 | 59.3 | 48.7 | 25.1 |
| | LWM (Liu et al., 2024e) | 7B | 75.2 | - | 55.8 | 44.8 | - |
| | Chameleon [†] | 7B | 77.8 | 1056.9 | 57.8 | 49.6 | 26.7 |
| | Orthus (Ours) | 7B | 79.6 | 1265.8 | <u>63.2</u> | 52.8 | 28.2 |

| Model | Res. | GenEval ↑ | HPSv2↑ | POPE↑ | MME↑ | GQA↑ |
|--------|------|-----------|--------|-------|--------|------|
| Orthus | 512 | 0.58 | 28.2 | 79.6 | 1265.8 | 52.8 |
| VILA-U | 256 | 0.40 | 25.3 | 83.9 | 1336.2 | 58.3 |
| Janus | 384 | 0.61 | 27.8 | 87.0 | 1338.0 | 59.1 |

Orthus: 文生图可视化结果

Show-o

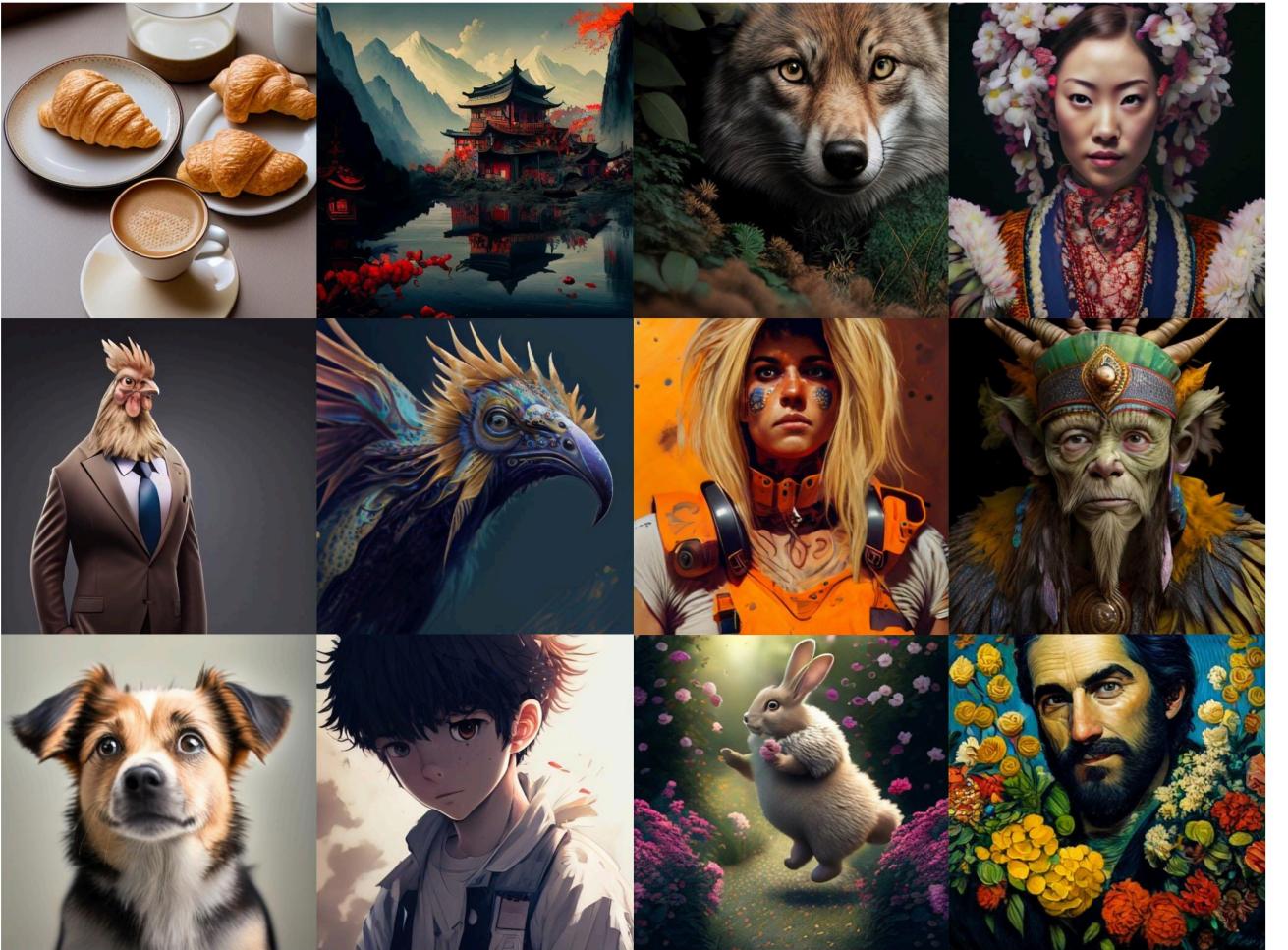


A detailed ink illustration of a hedgehog.

Chameleon



Orthus



A hyena fursona sits in a savannah sunset amidst the grass.



Oil painting portrait of a young woman in a field of flowers at sunset with mountains in the background.

Orthus: 图文交错生成结果 (图文->图、demo+图->图、图文->图文)

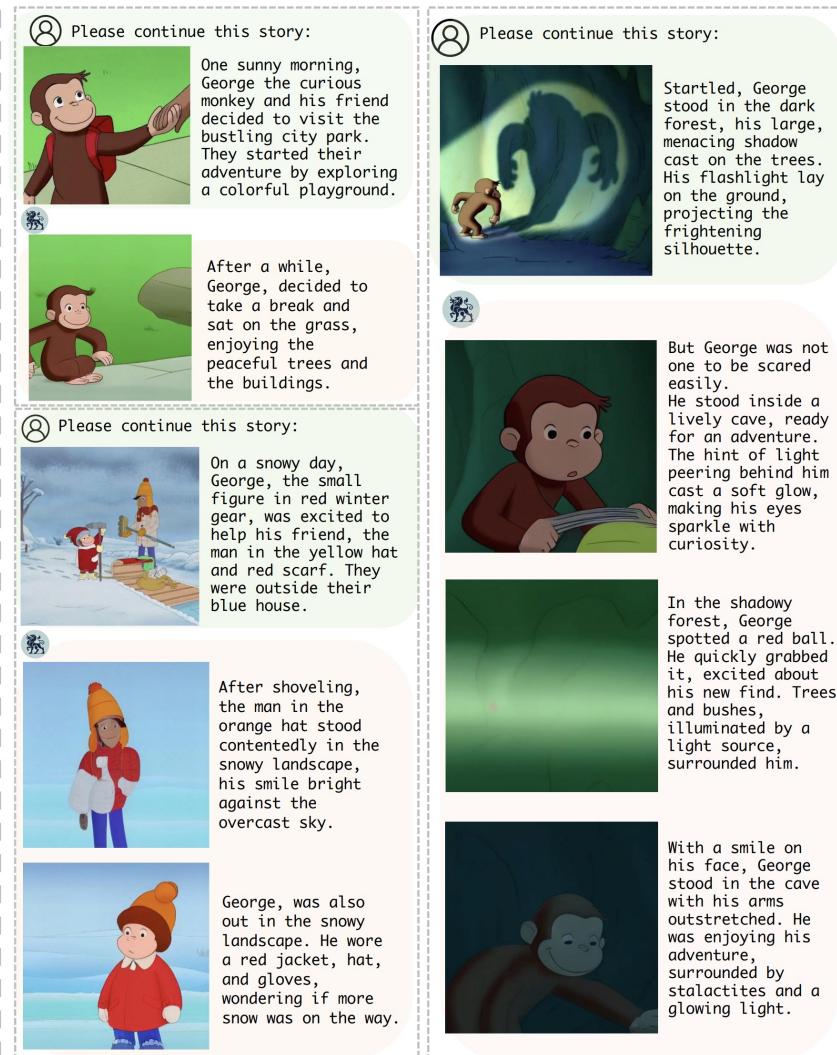


Table 1. Comparisons of CLIP similarities (Ruiz et al., 2023; Gal et al., 2022) between editing-specific diffusion models and Orthus on the test dataset of Instruct-Pix2Pix.

| Model | -T↑ | -I↑ | -D↑ |
|---------------------------------|--------------|-------------|--------------|
| PnP (Tumanyan et al., 2023) | 0.156 | 0.76 | 0.023 |
| SDEdit (Meng et al.) | 0.229 | 0.84 | 0.047 |
| I-Pix2Pix (Brooks et al., 2023) | 0.233 | 0.88 | 0.045 |
| Orthus (Ours) | 0.238 | 0.87 | 0.049 |

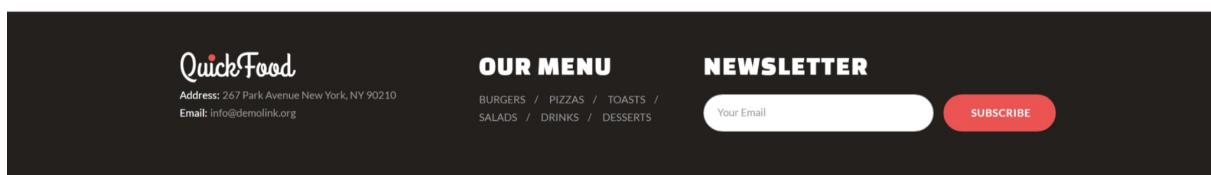
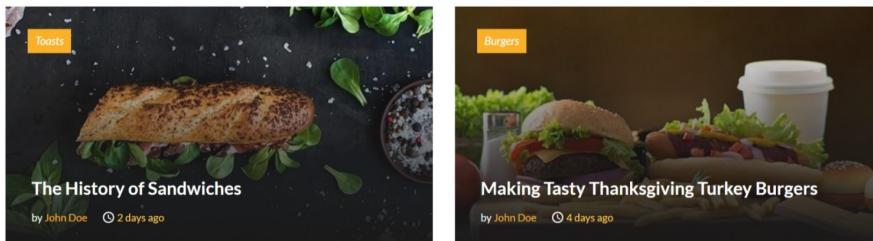
图像编辑能力指标

| | Orthus | Show-o | NExT-GPT | MiniGPT-5 | GILL | SEED-X |
|---------------|--------|--------|----------|-----------|------|--------|
| OpenING-IVD ↑ | 6.3 | 5.1 | 5.2 | 5.3 | 6.2 | 8.0 |

图文交错生成指标



Orthus: 图文交错的HTML网页生成



思路应用于阶跃星辰NextStep-1多模态模型

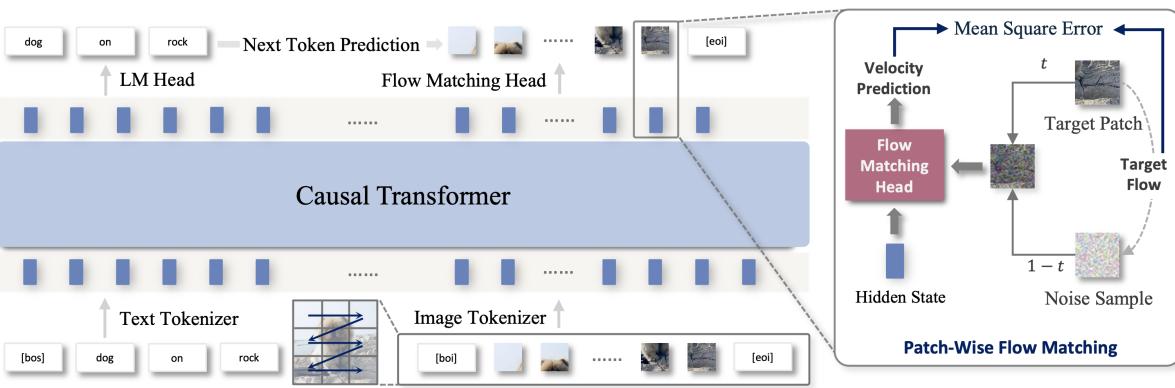


Figure 2 | Overview of NextStep-1 Framework. NextStep-1 employs a causal transformer to process tokenized text and image tokens. During training, Flow Matching Head predicts the continuous flow from a noise sample to the next target image patch, conditioned on the output hidden state. At inference, this allows for generating images by iteratively guiding noise to create the next patch.

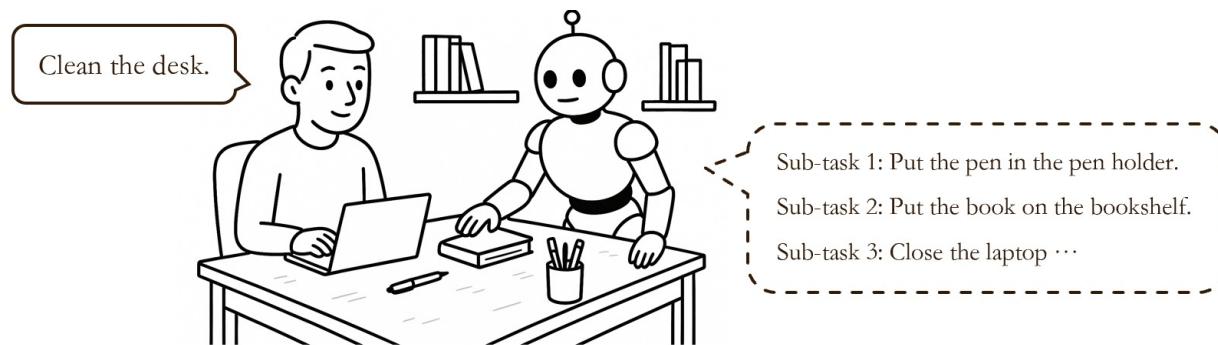
Table 2 | Comparison of image-text alignment on GenEval (Ghosh et al., 2023), GenAI-Bench (Lin et al., 2024), and DPG-Bench (Hu et al., 2024). * result is with rewriting. † result is with Self-CoT.

| Method | GenEval↑ | | GenAI-Bench↑ | | DPG-Bench↑ |
|---|------------------------|------------------------|------------------------|----------|------------|
| | Basic | Advanced | Basic | Advanced | |
| Proprietary | | | | | |
| DALL-E 3 (Betker et al., 2023) | 0.67 | - | 0.90 | 0.70 | 83.50 |
| Seedream 3.0 (Gao et al., 2025) | 0.84 | - | - | - | 88.27 |
| GPT4o (OpenAI, 2025b) | 0.84 | - | - | - | 85.15 |
| Diffusion | | | | | |
| Stable Diffusion 1.5 (Rombach et al., 2022) | 0.43 | - | - | - | - |
| Stable Diffusion XL (Podell et al., 2024) | 0.55 | 0.83 | 0.63 | - | 74.65 |
| Stable Diffusion 3 Medium (Esser et al., 2024) | 0.74 | 0.88 | 0.65 | - | 84.08 |
| Stable Diffusion 3.5 Large (Esser et al., 2024) | 0.71 | 0.88 | 0.66 | - | 83.38 |
| PixArt-Alpha (Chen et al., 2024b) | 0.48 | - | - | - | 71.11 |
| Flux-1-dev (Labs, 2024) | 0.66 | 0.86 | 0.65 | - | 83.79 |
| Transfusion (Zhou et al., 2025) | 0.63 | - | - | - | - |
| CogView4 (Z.ai, 2025) | 0.73 | - | - | - | 85.13 |
| Lumina-Image 2.0 (Qin et al., 2025) | 0.73 | - | - | - | 87.20 |
| HiDream-I1-Full (Cai et al., 2025) | 0.83 | 0.91 | 0.66 | - | 85.89 |
| Mogao (Liao et al., 2025) | 0.89 | - | 0.68 | - | 84.33 |
| BAGEL (Deng et al., 2025) | 0.82/0.88 [†] | 0.89/0.86 [†] | 0.69/0.75 [†] | - | 85.07 |
| Show-o2-7B (Xie et al., 2025b) | 0.76 | - | - | - | 86.14 |
| OmniGen2 (Wu et al., 2025b) | 0.80/0.86* | - | - | - | 83.57 |
| Qwen-Image (Wu et al., 2025a) | 0.87 | - | - | - | 88.32 |
| AutoRegressive | | | | | |
| SEED-X (Ge et al., 2024) | 0.49 | 0.86 | 0.70 | - | - |
| Show-o (Xie et al., 2024) | 0.53 | 0.70 | 0.60 | - | - |
| VILA-U (Wu et al., 2024) | - | 0.76 | 0.64 | - | - |
| Emu3 (Wang et al., 2024b) | 0.54/0.65* | 0.78 | 0.60 | - | 80.60 |
| SimpleAR (Wang et al., 2025c) | 0.63 | - | - | - | 81.97 |
| Fluid (Fan et al., 2024) | 0.69 | - | - | - | - |
| Infinity (Han et al., 2025) | 0.79 | - | - | - | 86.60 |
| Janus-Pro-7B (Chen et al., 2025b) | 0.80 | 0.86 | 0.66 | - | 84.19 |
| Token-Shuffle (Ma et al., 2025b) | 0.62 | 0.78 | 0.67 | - | - |
| NextStep-1 | 0.63/0.73 [†] | 0.88/0.90 [†] | 0.67/0.74 [†] | - | 85.28 |

文生图上的SOTA

多模态内容生成在文/图数据外的应用

- Vision-language-action (VLA)!
- VLA为什么需要离散-连续信号交错生成?
 - long-horizon任务：高层次目标，需要多个步骤的解决方案

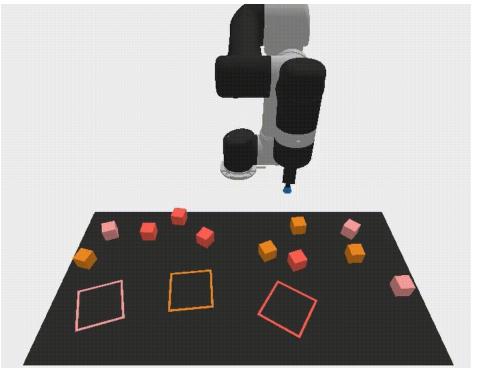


- 要求兼顾高层任务规划（目标->子任务）与低层动作控制（子任务->动作）
- 子任务（文本/图像）和动作（末端执行器位置、夹持器开合）模态不同

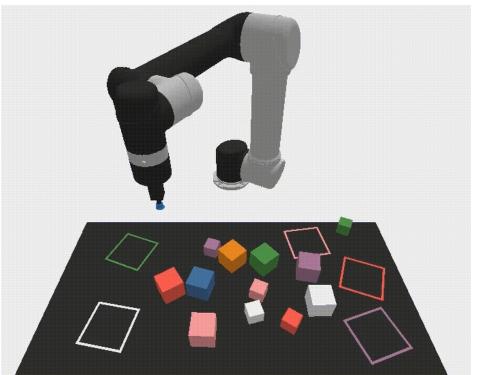
LoHoVLA: 显著提高模型的规划能力和未见任务的泛化性能

Table 2: Comparison of the average award (%) and success rate (%) on LoHoRavens benchmark. Bold entries indicate the highest success rates, underlined entries indicate the second-highest.

| Tasks | Vanilla VLA | LoHoRavens | | LoHoVLA |
|--------------|-------------|--------------------|-------------------|--------------------|
| | | Explicit feedback | Implicit feedback | |
| Seen Tasks | A | 79.0 / 79.0 | 67.3 / - | 67.3 / - |
| | B | 14.9 / 0.0 | 31.4 / - | 97.8 / 91.5 |
| | C | <u>26.8</u> / 0.5 | 18.0 / - | 34.9 / 22.5 |
| | D | 32.3 / 3.0 | 30.4 / - | 35.8 / 11.5 |
| | E | <u>22.1</u> / 3.5 | 9.6 / - | 85.1 / 81.0 |
| Unseen Tasks | F | <u>52.1</u> / 9.0 | 28.5 / - | 86.1 / 41.0 |
| | G | 6.8 / 0.0 | <u>21.9</u> / - | 40.1 / 25.0 |
| | H | 7.3 / 0.0 | <u>13.2</u> / - | 16.7 / 7.5 |
| | I | <u>43.1</u> / 1.5 | 12.8 / - | 77.2 / 52.0 |
| | J | <u>38.6</u> / 10.5 | 27.4 / - | 43.6 / 22.0 |
| | K | <u>58.2</u> / 33.0 | 4.0 / - | 73.8 / 54.5 |



"Move all blocks of a color that occur in even numbers to the same colored zone."

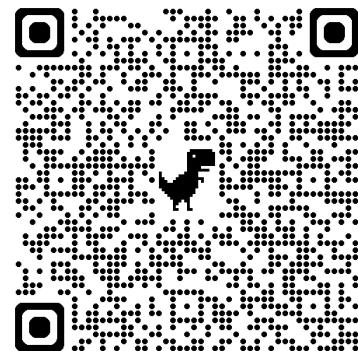


"Stack blocks of the same color in the zone with same color, with bigger blocks underneath."

感谢各位专家！敬请批评指正！

邮箱：zhijied@sjtu.edu.cn

主页：<https://thudzj.github.io/>



小红书



知乎



GitHub